



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS,  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/717,680	11/21/2000	Martyn S. Lovell	777.334US1	9114

23377 7590 04/21/2004

WOODCOCK WASHBURN LLP  
ONE LIBERTY PLACE, 46TH FLOOR  
1650 MARKET STREET  
PHILADELPHIA, PA 19103

EXAMINER
----------

SHRADER, LAWRENCE J

ART UNIT	PAPER NUMBER
----------	--------------

2124

DATE MAILED: 04/21/2004

9

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/717,680

Applicant(s)

LOVELL ET AL.

Examiner

Lawrence Shrader

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 15 January 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-28 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-28 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

Art Unit: 2124

### **DETAILED ACTION**

1. This action is in response to the amendment filed on January 15, 2004.
2. The arguments of the amendment of January 15, 2004 have been considered, but are moot in view of new applied art.

### ***Specification***

3. The objection to the Abstract exceeding 150 words in length is withdrawn in view of the amendment.

### ***Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1, 2, 8 – 10; 15 – 20; and 22 – 27 are rejected under 35 U.S.C. 102(e) as being anticipated by Charisius et al., U.S. Patent Application Publication US 2002/0016954 (hereinafter referred to as Charisius).

**In regard to claim 1, Charisius discloses a visual programming environment:**

Art Unit: 2124

*“A source code editor...;”*

A source code editor is disclosed (para. [0125])

*“A graphical design surface operable to display a graphical object representing actual code of the source code module;”*

A graphical display window (design surface) of actual source code of the source code module is disclosed (para. [0125]; e.g., Figure 27 shows a graphical object representing the actual source code).

*“...upon a change in the source code module, the change in the source code module is immediately communicated to the graphical design surface...”*

A source code edit is reflected in the graphical window (para. [0125]; e.g., Figure 27).

**In regard to claim 2**, incorporating the rejection of claim 1:

*“...a change in the graphical design surface is immediately communicated to the source code editor...”*

Modifying the object module results in an update to the source code (para. [0125; Figures 27 – 30).

**In regard to claim 8**, incorporating the rejection of claim 1:

*“...comprising at least one compiler...”*

Charisius specifies a compiler in paragraph [0061].

**In regard to claim 9**, incorporating the rejection of claim 1:

*“...the design surface is operative to bind the source code module to at last one compiler...”*

Charisius discloses a compile function that interacts with the graphical window in paragraph [0061]. It would be inherent that a software programming environment would necessarily bind the code to the compiler in order to obtain executable code.

**In regard to claim 10, incorporating the rejection of claim 1:**

*“...the design surface displays a graphical object...”*

Charisius discloses a design surface displaying a graphical object representing a database object (e.g., Figures 27 and 30).

**In regard to claim 15, Charisius discloses a visual programming environment:**

*“Creating a graphical object on a design surface, the graphical object representing actual code of a software module;”*

Actual code of a software module is displayed of an object module is provided (para. [0125]; e.g., Figure 27 shows a graphical object representing the actual source code).

*“Generating source code particular to the application type.”*

When the object model is modified, the code is modified according to the type of data structure (para. [0125]; e.g., Figure 27 shows a graphical object representing the actual source code).

*“Binding the graphical object to an application type;”*

Charisius discloses binding of statements to a database application (para. [0125]).

**In regard to claim 16, incorporating the rejection of claim 15:**

*“...wherein the application type is a source code compiler.”*

The Charisius invention specifies source code for compilation by a compiler (column 5, lines 32 – 36).

**In regard to claim 17, incorporating the rejection of claim 15:**

*“...wherein the application type is a database application.”*

Art Unit: 2124

The Charisius invention discloses binding of statements to a database application (para. [0125]).

**In regard to claim 18**, incorporating the rejection of claim 15:

*“...wherein the application type is a source code interpreter.”*

The Charisius invention does not explicitly reference binding the object to an interpreter, but a compiler is specified at paragraph [0061]. Therefore, it would have been obvious to one skilled in the art that an interpreter could have been specified to “compile” the code line-by-line in order to execute the code.

**In regard to claim 19**, incorporating the rejection of claim 15:

*“Modifying the source code; and”*

*“Refreshing the design surface to update the graphical object to reflect the modification to the source code.”*

Source code editing is reflected in the graphical window (para. [0125]; see, e.g., Figures 27 and 30).

**In regard to claim 20**, incorporating the rejection of claim 15:

*“Modifying the graphical object on the design surface; and”*

*“Refreshing the source code to reflect the modification to the graphical object.”*

Charisius modifies the object module and the modification results in an update to the source code (para. [0125]; e.g., Figures 27 and 30).

**In regard to claims 22 - 27** (the computer-readable medium), they are rejected for the corresponding reasons put forth in the rejection of claims 15 – 20 (the method).

***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 3 – 7; 21; and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Charisius et al., U.S. Patent Application Publication US 2002/0016954, in view of Washburn et al., U.S. Patent 5,157,779 (hereinafter referred to as Washburn).

**In regard to claims 3, incorporating the rejection of claim 1:**

*“A change manager operative to manage versioning...;”*

Charisius discloses a visual programming environment using a text editor to enter and modify source code, but does not teach a version manager. However, Washburn discloses a compare module, which manages two versions of a file (column 2, lines 5 – 8; e.g., Figure 20). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify visual programming environment in the disclosure of Charisius with the compare manager as taught by Washburn because the compare manager provides the capability to pull up a previous version to determine the differences in the code, restore old code, modify older versions, etc. in order to save programming time.

*“An application data store operative to store a previous version...”*

Charisius discloses a visual programming environment using a text editor to enter and modify source code, but does not teach a storage means for previous versions. However,

Art Unit: 2124

Washburn discloses a data store module, which stores the master file (the previous version) of the file (column 2, lines 3 – 5; e.g., Figure 20). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify visual programming environment in the disclosure of Charisius with the data store module as taught by Washburn because the compare manager needs previous versions to work with, and an obvious means to store them would be necessary.

**In regard to claims 4 and 7, incorporating the rejection of claim 3:**

*“..the difference between the source code module and the previous version of the source code module is highlighted...”*

Charisius discloses a visual programming environment using a text editor to enter and modify source code, but does not teach highlighting a difference between the source code module and the previous version of the module. However, Washburn teaches highlighting textual differences in a text editor, as in claim 4 (column 12, lines 6 – 8), and highlighting differences graphically in the graphics window, as in claim 7 (column 10, lines 52 – 59; e.g., Figure 7d). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the source code editor and the graphical design surface in the teaching of Charisius with the text difference highlighting feature and the graphical difference highlighting feature as taught by Washburn because it would allow the user to efficiently determine at a glance the differences between two files, e.g., highlighting differences between files of different versions.

**In regard to claim 5 and 6, incorporating the rejection of claim 4.**

*“...the difference is highlighted using...”*



Charisius does not teach highlighting a difference between the source code module and the previous version in the code editor with either a squiggly line (claim 5) or a tooltip bar (claim 6) to highlight the difference. Washburn does teach highlighting textual differences (column 12, lines 6 – 8), but does not specify the type of highlighting. However, official notice is taken that highlighting a segment of text is well known in the art and can be done in many ways, e.g., color change of text or background, font change, italicization, bolding, tooltip, squiggly underlines, straight underlines, etc. to name a few. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the source code editor and the graphical design surface in the teaching of Charisius with the text difference highlighting feature and the graphical difference highlighting feature as taught by Washburn combined with the well known knowledge of various different types of highlighting, because it would allow the user to efficiently determine at a glance the differences between two files, e.g., highlighting differences between files of different versions.

**In regard to claim 21, incorporating the rejection of claim 15:**

*“...reading a template having a pre-configured software module from a datastore.”*

Charisius discloses a visual programming environment using a text editor to enter and modify source code, but does not teach a storage means for previous versions. However, Washburn discloses a data store module, which stores the master file (the previous version) of the file (column 2, lines 3 – 5; e.g., Figure 20). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify visual programming environment in the disclosure of Charisius with the data store module as taught by Washburn because the

Art Unit: 2124

compare manager needs previous versions to work with, and an obvious means to store them would be necessary.

**In regard to claim 28** (the computer-readable medium), incorporating the rejection of claim 22, rejected for the corresponding reason given in the rejection of claim 21 (the method).

8. Claims 11 – 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Charisius et al., U.S. Patent Application Publication US 2002/0016954 as applied to claim 10 above, in view of Williamson et al., U.S. Patent 6,122,641 (hereinafter referred to as Williamson), and further in view of Peddada et al., U.S. Patent 6,031,533 (hereinafter referred to as Peddada) .

**In regard to claim 11**, incorporating the rejection of claim 10:

*“...the database object further includes a database column, wherein the source code module includes a variable, wherein the design surface is operative to bind the database column to the variable.”*

Charisius discloses a design surface displaying a graphical object representing a database object, but does not teach the implementation of a database column bound to a variable.

However, Williamson teaches mapping data to fetched objects in a model by binding a variable to a database column (column 15, lines 31 – 41). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Charisius invention with the binding of database columns to a variable as taught by Williamson, because the modification provides the ability to efficiently map properties of the objects from the columns of the database to the displayed objects.

**In regard to claim 12**, incorporating the rejection of claim 11:

*“...the binding is established through a drag-and-drop interface.”*

Neither Charisius nor Williamson teaches the use of a drag-and drop interface. However, Peddada teaches a drag-and-drop interface to bind a graphics object to a program. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to implement the visual programming environment of Charisius to display graphical objects from a database, modified by Williamson binding a variable to a database column, further modified by Peddada to provide a drag-and-drop feature to accomplish the added functions enabled by the above combinations, because the drag-and-drop feature further simplifies the programming function for users with little or no programming experience.

9. **Claim 13** is rejected under 35 U.S.C. 103(a) as being unpatentable over Charisius et al., U.S. Patent Application Publication US 2002/0016954 as applied to claim 1 above, in view of Gupta et al., U.S. Patent 6,484,156 (hereinafter referred to as Gupta)

*“...provide an interface to highlight a set of software modules that are grouped together as a package.”*

Charisius does not disclose highlighting a set of modules grouped together. However, Gupta discloses highlighting a set or sets of annotations (modules) for execution (column 14, lines 59 – 67). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to enhance the visual programming environment taught by Charisius with the feature of highlighting a set of modules to be downloaded as taught by Gupta because the combination allows the user to select and run a set of modules via the graphics window without needing any detailed programming knowledge.

10. **Claim 14** is rejected under 35 U.S.C. 103(a) as being unpatentable over Charisius et al., U.S. Patent Application Publication US 2002/0016954, in view of Gupta et al., U.S. Patent 6,484,156 (hereinafter referred to as Gupta) as applied to claim 13, and further in view of O'Donnell et al., U.S. Patent 6,223,203 (hereinafter referred to as O'Donnell)

*“...receive a list of system identifiers...identifying a particular computer system...”*

Charisius provides a visual programming interface, but neither Charisius nor Gupta discloses a means to receive a list of system identifiers of a particular computer system to deploy a package. However, O'Donnell discloses a means to receive a list of system identifiers of a particular computer system. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to enhance the visual programming environment taught by Charisius with the feature of highlighting a set of modules to be downloaded as taught by Gupta, and further modified with a received list of list of possible computer systems to select a particular system for module deployment as taught by O'Donnell, because this added feature provides the user a means to select a particular computer system to receive the software module discloses by the Gupta invention giving the user more detailed control over the system configuration.

#### ***Response to Arguments***

11. Applicant's arguments with respect to claim 1 – 28 have been considered but are moot in view of the new ground(s) of rejection.

Art Unit: 2124

*Conclusion*

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (703) 305-8046. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703) 305-9662. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Lawrence Shrader  
Examiner  
Art Unit 2124

April 6, 2004



**TODD INBERG**  
**PRIMARY EXAMINER**